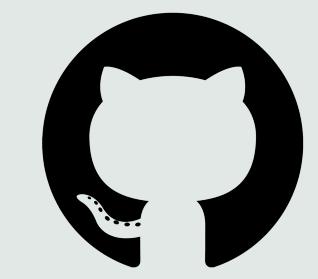


# An introduction to GitHub

Rose Nasrawi

03/11/23

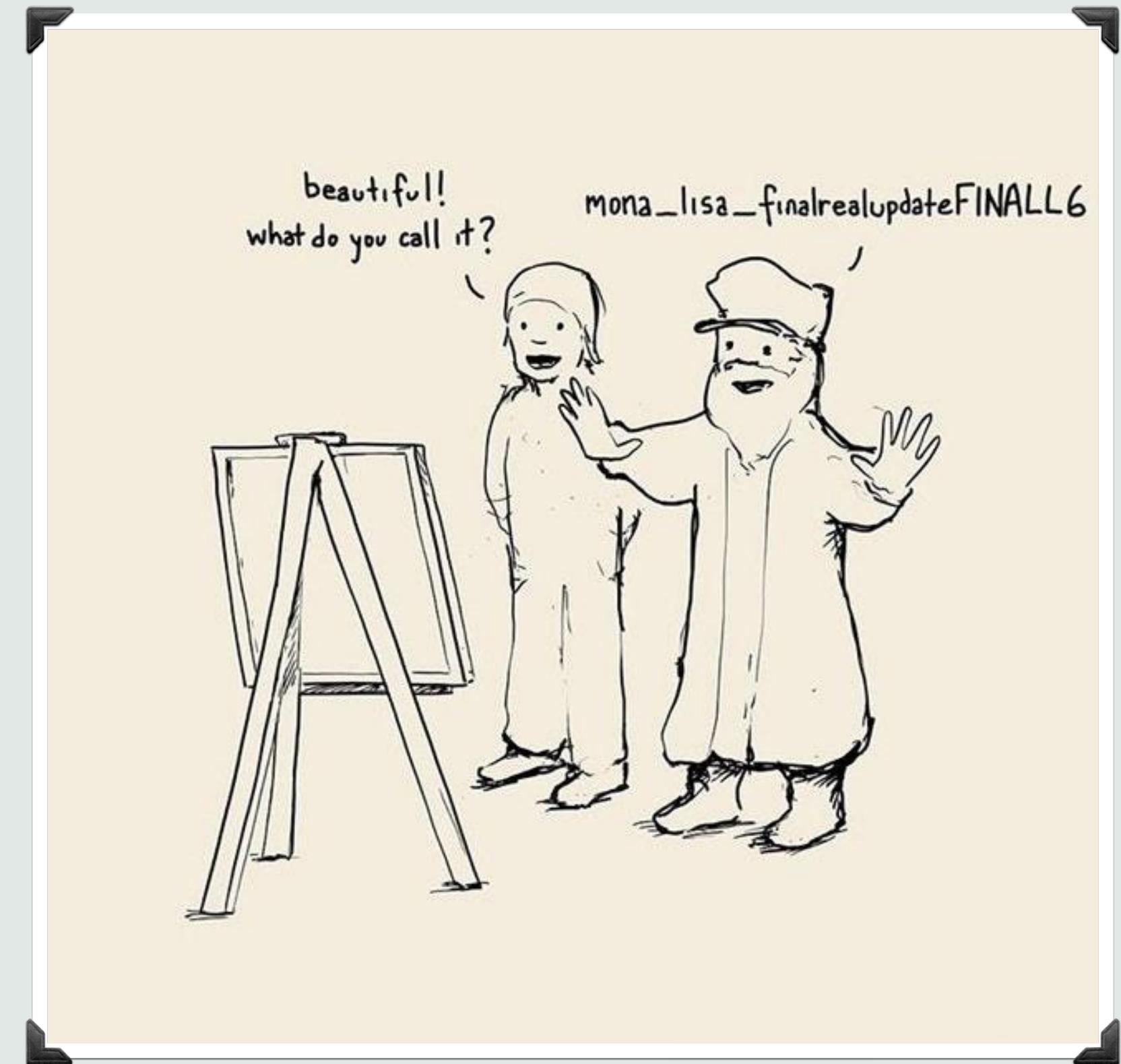


**GitHub**

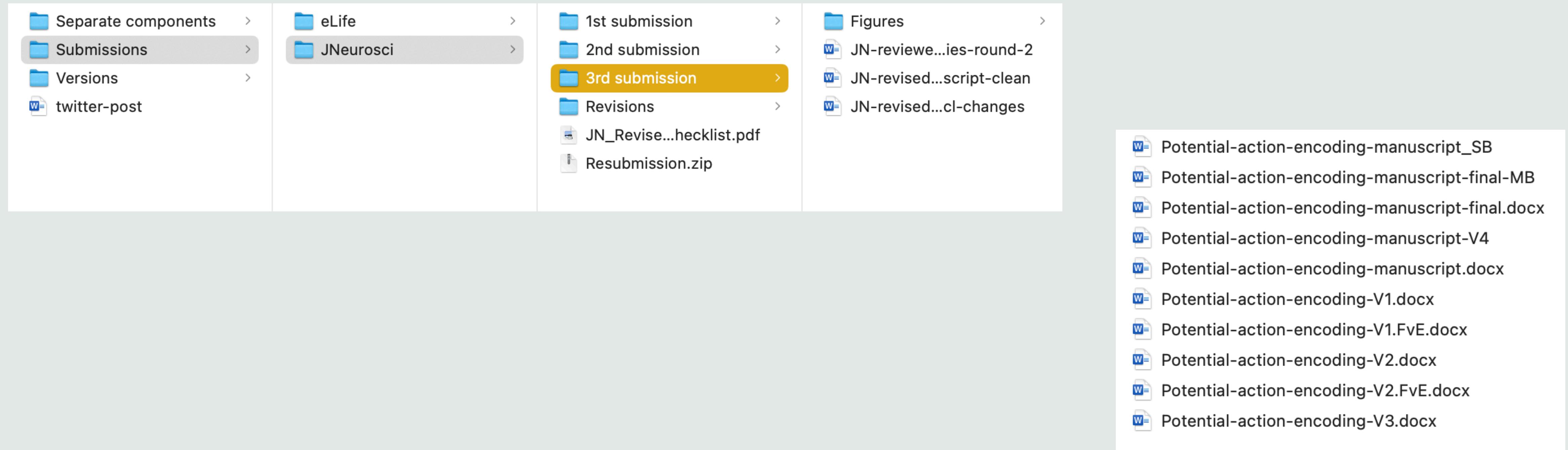
|

**what/why/how?**

# GitHub | Version control



# GitHub | Version control



This can quickly get out of hand!

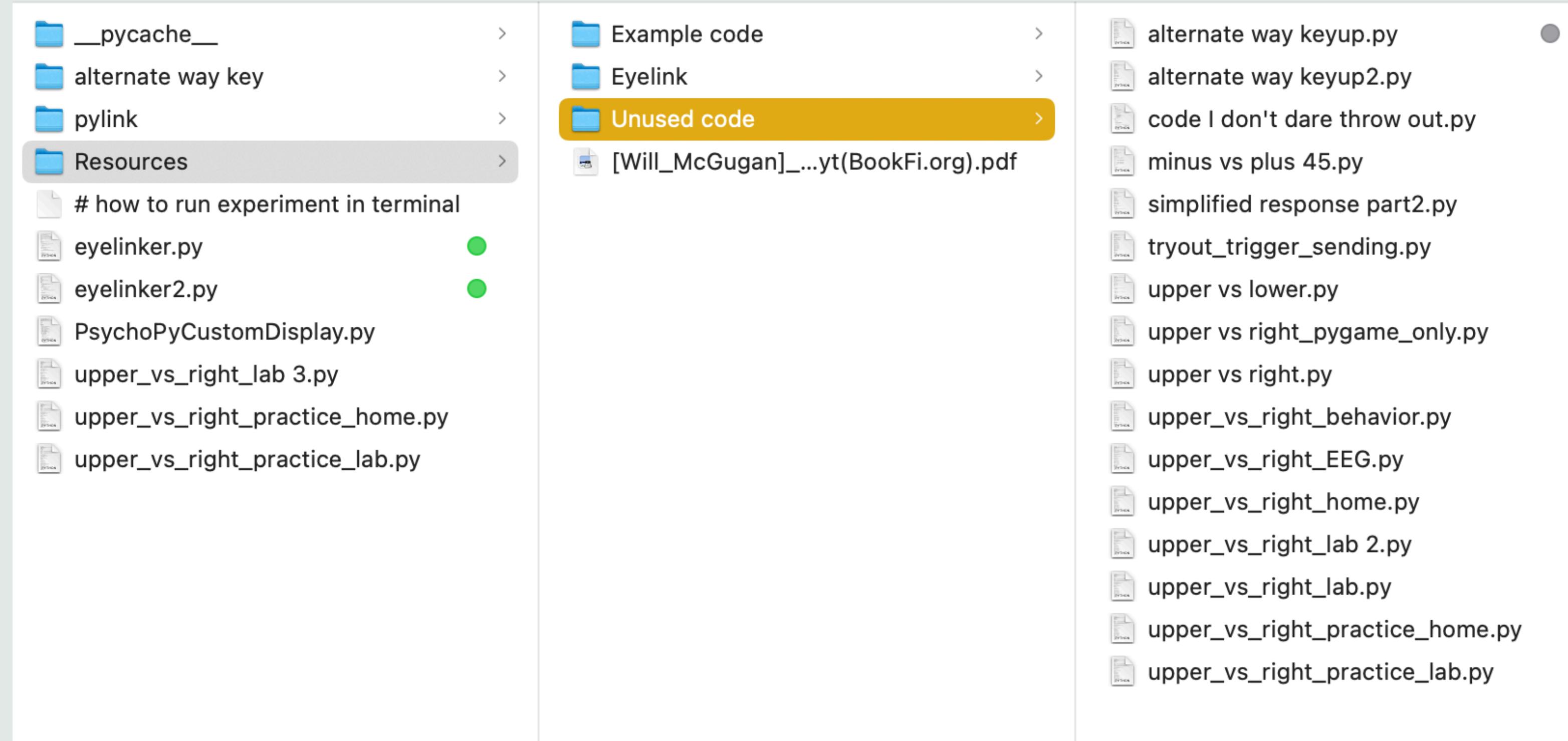


# Microsoft Word ‘Track-changes’



**Keeping track of the changes in your document**

# GitHub | Version control

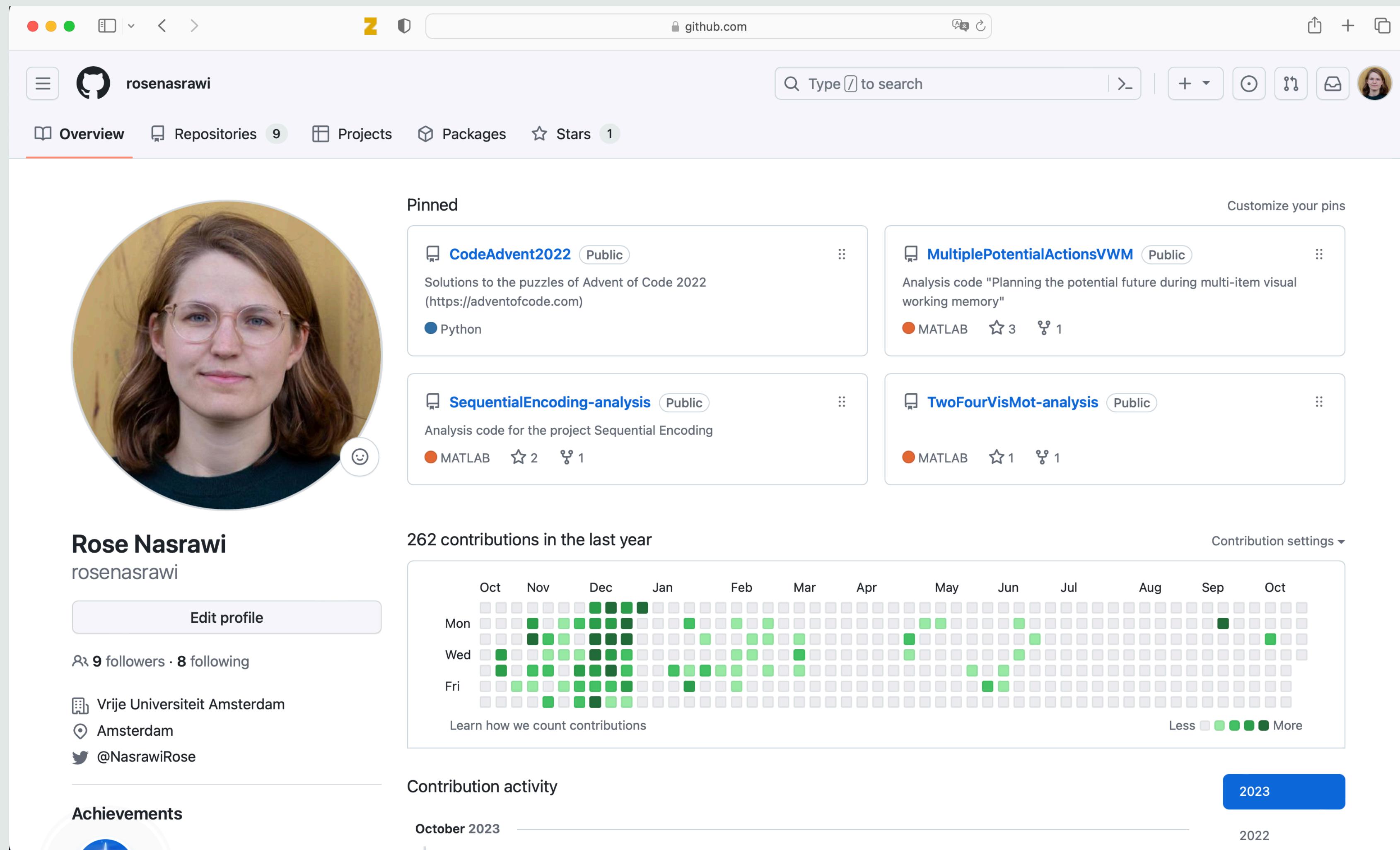


This can quickly get out of hand!



code I don't dare throw out.py

# GitHub to the rescue!



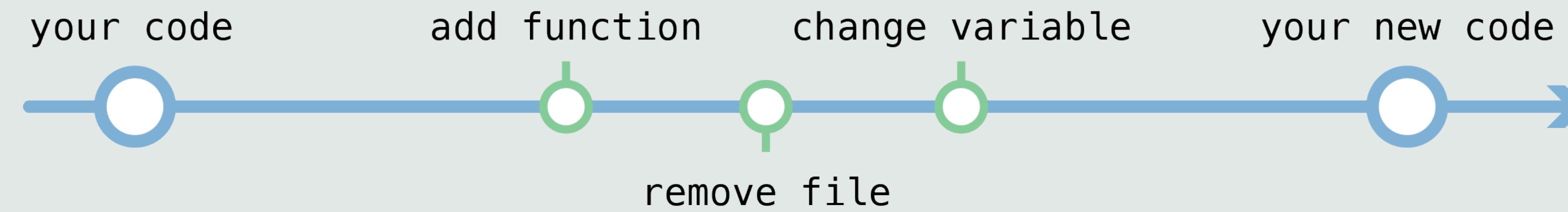
The screenshot shows a GitHub profile page for the user `rosenasrawi`. The profile picture is a circular portrait of a woman with brown hair and glasses. The top navigation bar includes links for Overview, Repositories (9), Projects, Packages, and Stars (1). The main content area features a pinned section with four repository cards:

- CodeAdvent2022** (Public) - Solutions to the puzzles of Advent of Code 2022 (<https://adventofcode.com>)
- MultiplePotentialActionsVWM** (Public) - Analysis code "Planning the potential future during multi-item visual working memory"
- SequentialEncoding-analysis** (Public) - Analysis code for the project Sequential Encoding
- TwoFourVisMot-analysis** (Public) - Analysis code for the project TwoFourVisMot

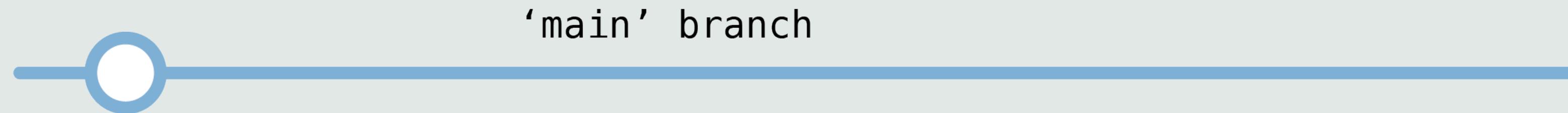
Below the pinned section, there is a heatmap showing contribution activity for the last year. The heatmap grid spans from October of the previous year to October of the current year, with days of the week on the y-axis and months on the x-axis. Darker squares indicate higher contribution frequency. A legend at the bottom right of the heatmap shows color-coded scales for 'Less' (light gray) and 'More' (dark green).

On the left side of the profile page, there is a sidebar with the user's name, handle, follower count (9), following count (8), and links to their institutional profile (Vrije Universiteit Amsterdam) and social media (Twitter @NasrawiRose). There is also a section for achievements.

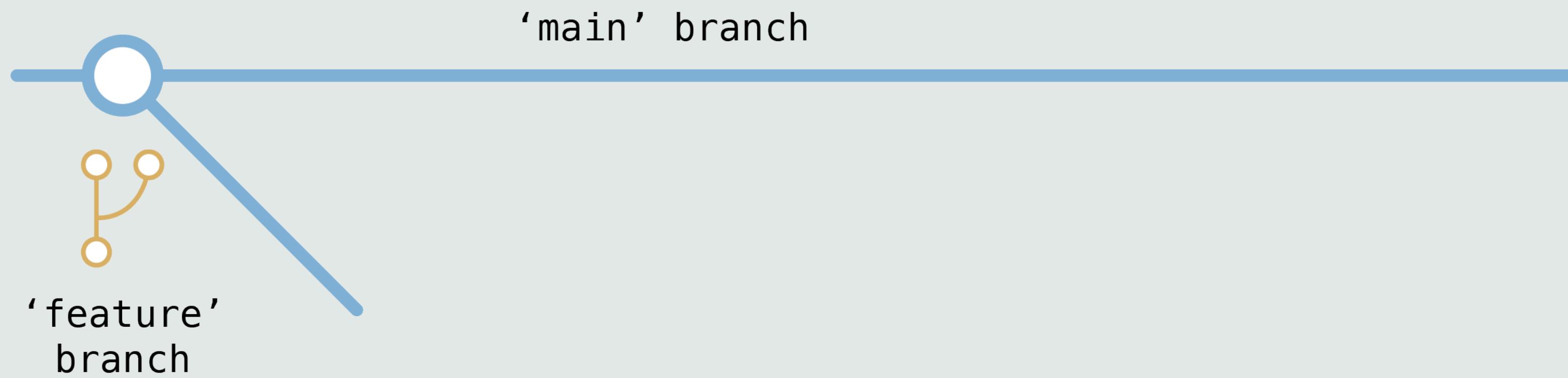
# GitHub | 'commit' history



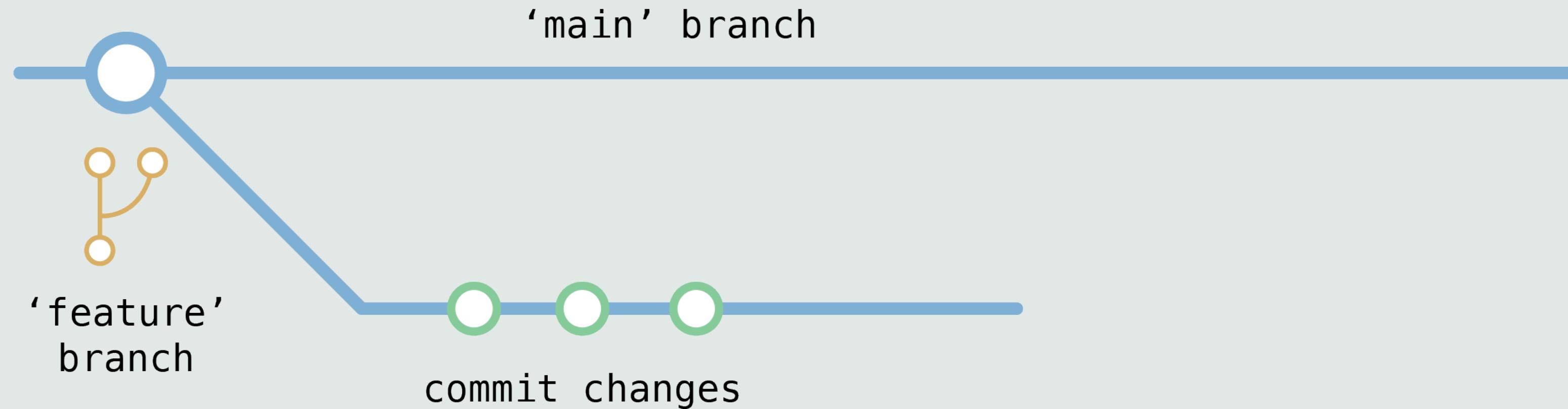
# GitHub | 'commit' history



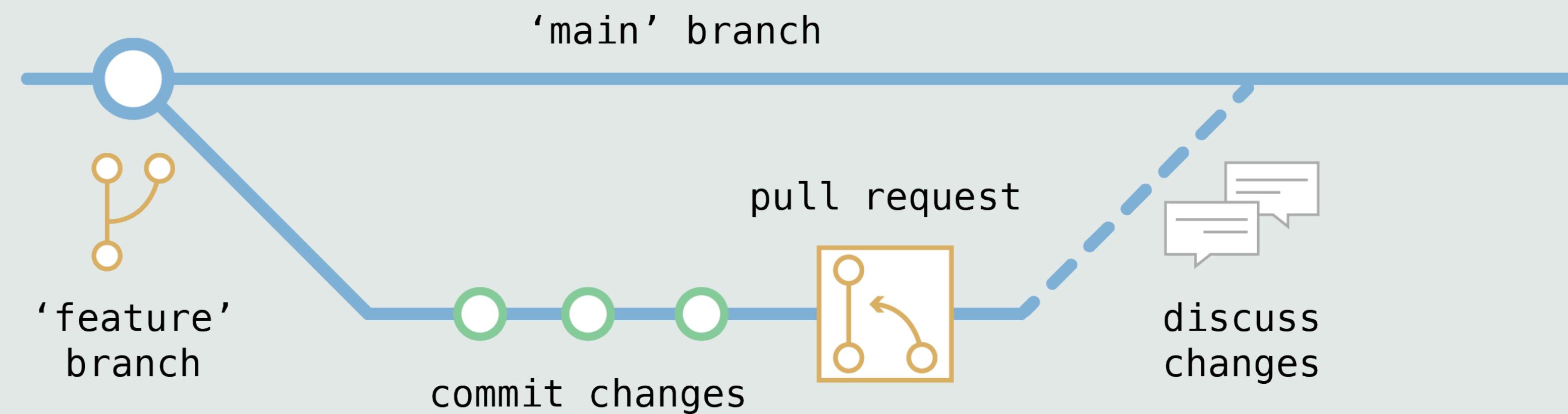
# GitHub | 'commit' history



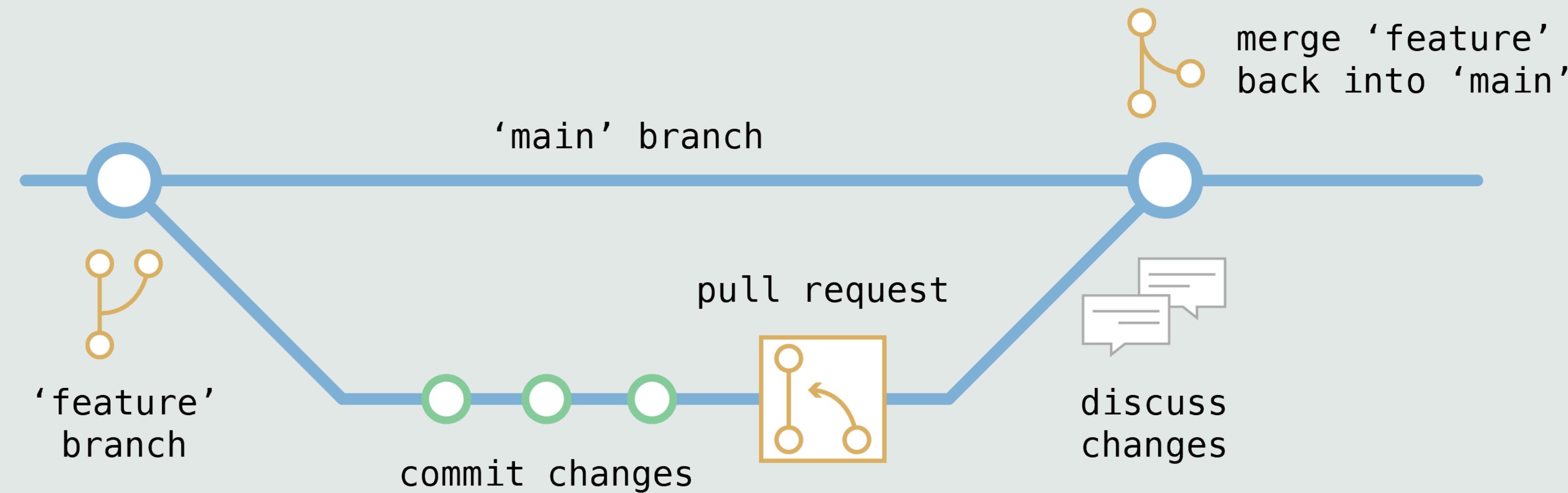
# GitHub | 'commit' history



# GitHub | 'commit' history



# GitHub | 'commit' history



# GitHub | Repository

The screenshot shows a GitHub repository page for 'CodeAdvent2022'. The repository is public and contains 1 branch and 0 tags. The main file listed is 'README.md', which contains the text: 'CodeAdvent2022' and 'My solutions to [Advent of Code 2022](#), using Python.' Below the README, there are status badges for 'day 25' and 'stars 50'. The repository has 141 commits from rosenasrawi, dated Dec 25, 2022. The commit history includes files like '.github/workflows', 'input', 'solutions', '.gitignore', and 'README.md'. The repository page also includes sections for 'About', 'Releases', 'Packages', and 'Languages'.

**Organise a project:**  
**folders**  
**files**  
**README**  
**.gitignore**

# GitHub | Repository

The screenshot shows a GitHub repository page for 'rosenasrawi / CodeAdvent2022'. The 'Code' tab is selected. The main area displays the commit history for December 25, 2022. A dropdown menu shows 'main' is selected. The commits are listed in chronological order from top to bottom:

- Update name day 22 by rosenasrawi committed on Dec 25, 2022. Commit hash: 172aad3.
- Update badges by rosenasrawi authored and github-actions[bot] committed on Dec 25, 2022. Commit hash: 6970ad4.
- Solve day 25 by rosenasrawi committed on Dec 25, 2022. Commit hash: 1cdde4c.
- Update badges by rosenasrawi authored and github-actions[bot] committed on Dec 25, 2022. Commit hash: 809bc2b.
- Solve day 24 by rosenasrawi committed on Dec 25, 2022. Commit hash: 78cf6e.
- Refactor day 23 by rosenasrawi committed on Dec 25, 2022. Commit hash: eaf36c8.
- Update badges by rosenasrawi authored and github-actions[bot] committed on Dec 25, 2022. Commit hash: 197b0c7.

Below this, there is a section for 'Commits on Dec 24, 2022' with one entry: 'Update badges' by rosenasrawi committed on Dec 24, 2022. Commit hash: f2d2200.

## 'Commit' history

The screenshot shows a GitHub commit history for a file named `solutions/day-23.py`. The commit is titled "Refactor day 23" and was made by `rosenasrawi` on Dec 25, 2022. It has 1 parent commit `197b0c7` and a commit hash `eaf36c8`. The commit message indicates 29 additions and 50 deletions. The code changes are shown in a diff view:

```
@@ -11,11 +11,9 @@ def findelves(input, elves = []):
11         elf = (x,y)
12         elves.append(elf)
13
14 -     return elves
15
16 - def getorder():
17 -
18 -     return [('N','NE','NW'), ('S','SE','SW'), ('W','NW','SW'),
19 - ('E','NE','SE')]
20
21     def adjacent(x,y):
22
@@ -28,85 +26,66 @@ def adjacent(x,y):
28
29     def elfupdate(elves, order):
30
31 -     moves = []
32 -     newelves = []
33 -
34     elfset = set(elves)
35
36 -     for elf in elves:
37         x,y = elf
38 -     adj = adjacent(x,y)
39         dirs = adj.values()
40
41 -     for i, dir in enumerate(elves):
42         if dir == adj:
43             moves.append((i, dirs))
44
45     for i, dir in enumerate(elves):
46         if dir in moves:
47             newelves.append(elves[i])
48
49     return newelves, order
50
51
52     def adjacent(x,y):
53
54         moves = []
55
56         for i, elf in enumerate(elves):
57             if (x,y) == elf:
58                 moves.append((i, adj))
59
60         return moves
61
62
63     def getorder():
64
65         return [('N','NE','NW'), ('S','SE','SW'), ('W','NW','SW'),
66 ('E','NE','SE')]
```

**'Commit' history**  
overview of all  
changes, per file

# GitHub | Repository

Remote

vs

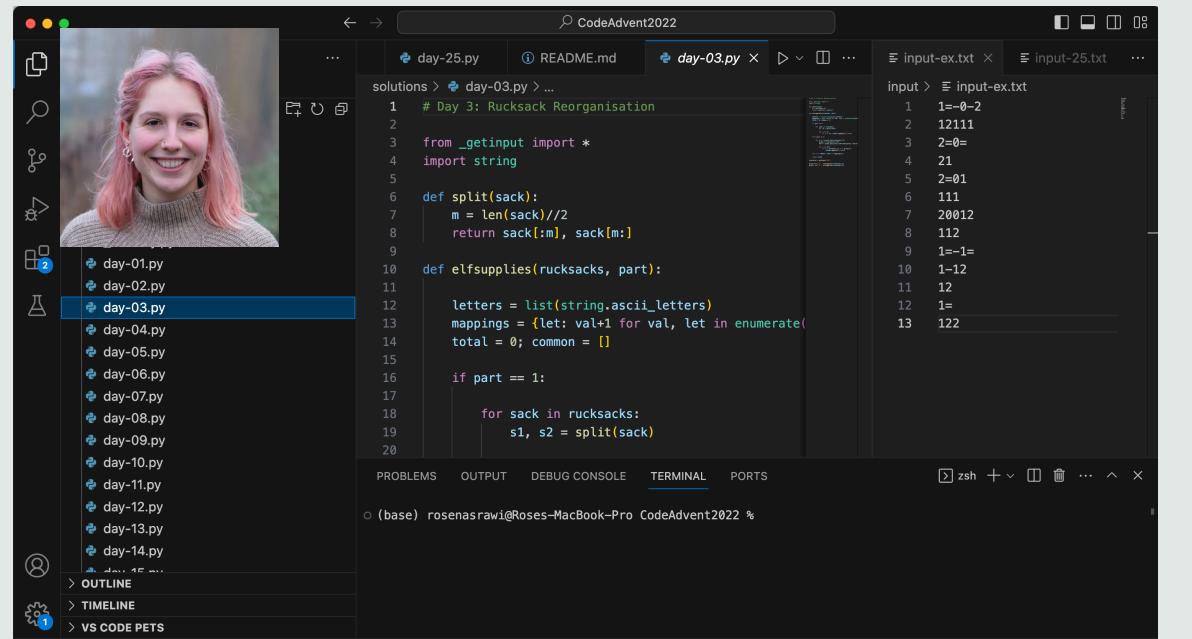
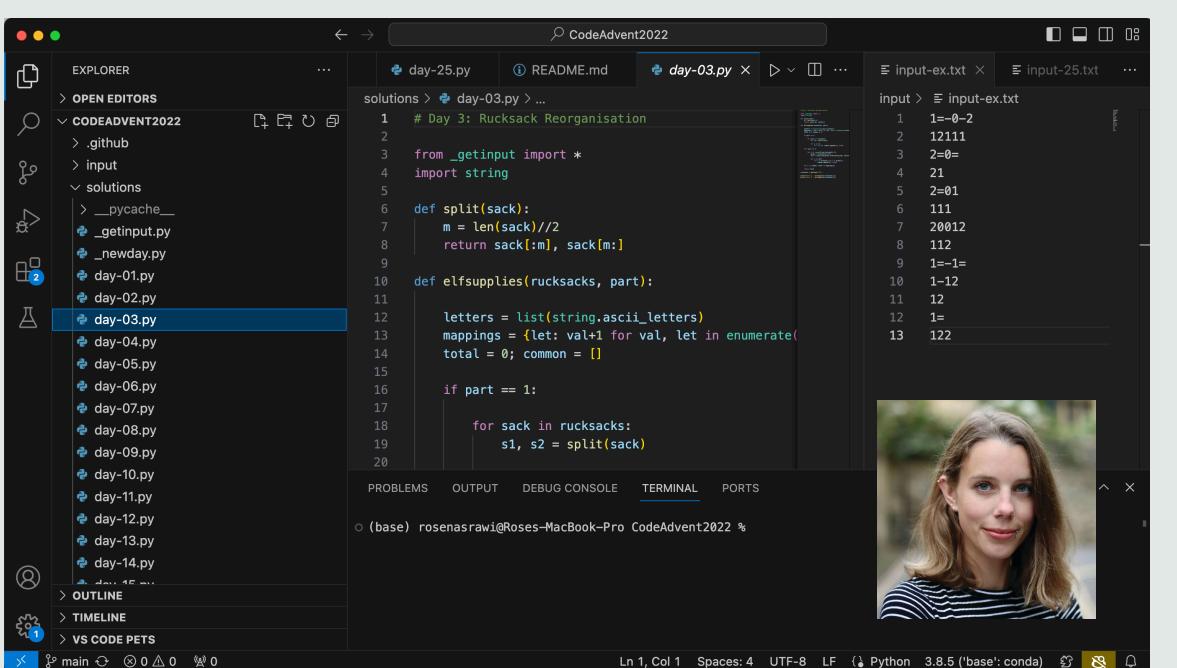
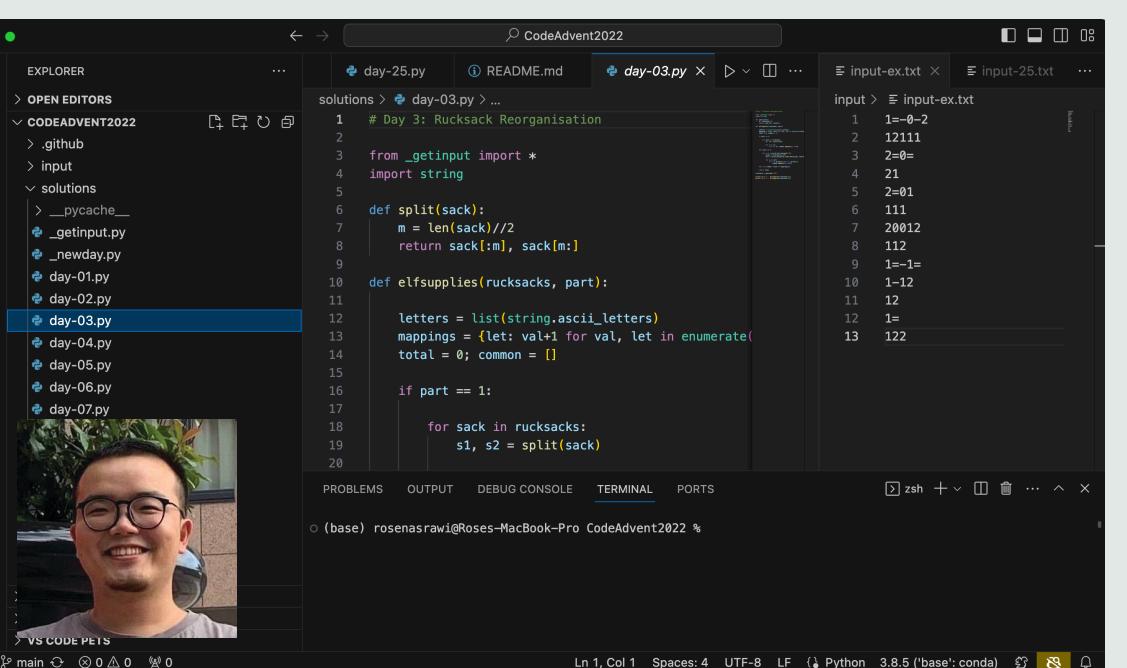
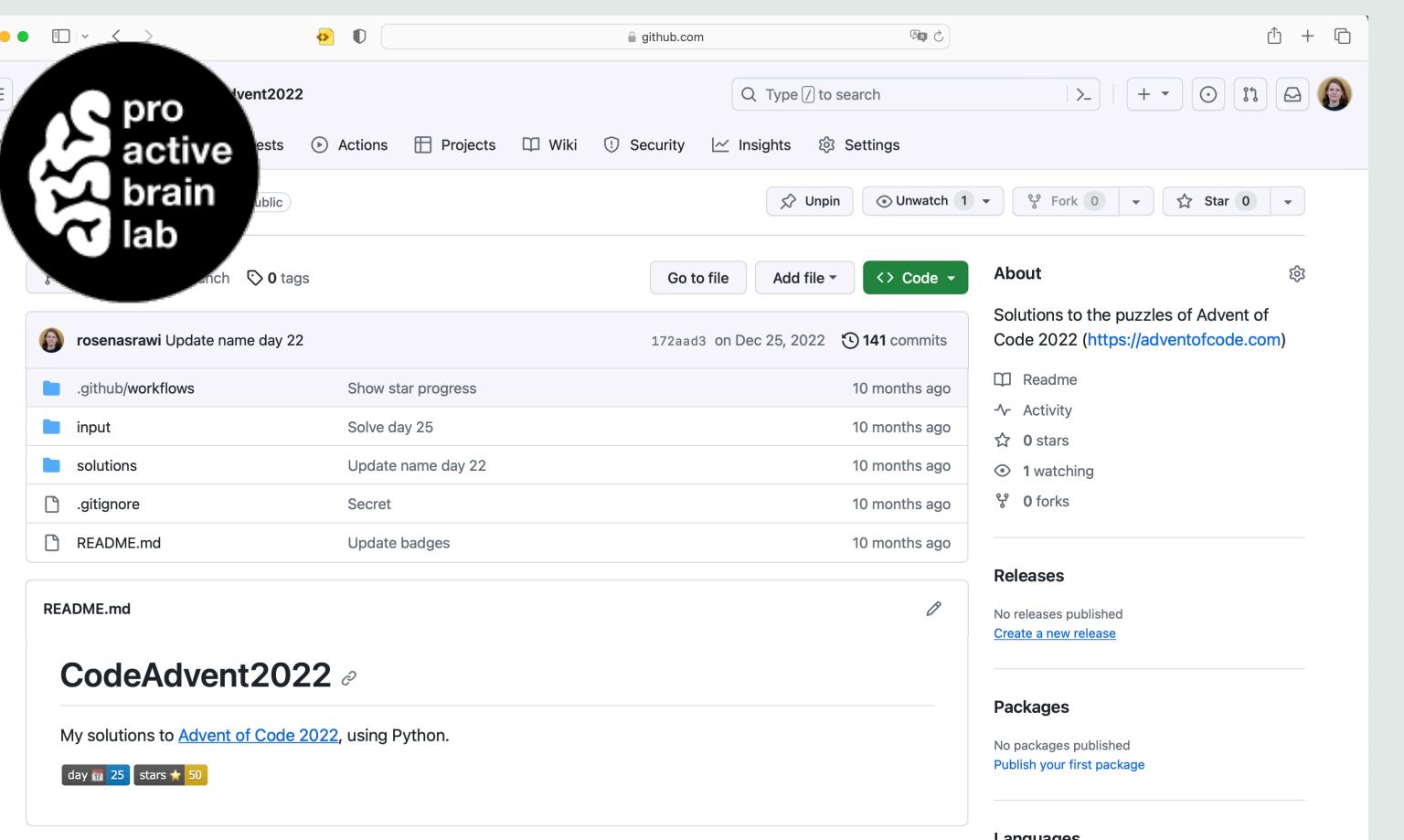
Local

The screenshot shows a GitHub repository page for 'CodeAdvent2022'. The repository is public and has 1 branch and 0 tags. It contains files like .github/workflows, input, solutions, .gitignore, and README.md. The README.md file describes the repository as containing solutions to the puzzles of Advent of Code 2022. The code is available at <https://adventofcode.com>. The repository has 141 commits from rosenasrawi, updated on Dec 25, 2022. The repository has 25 days and 50 stars.

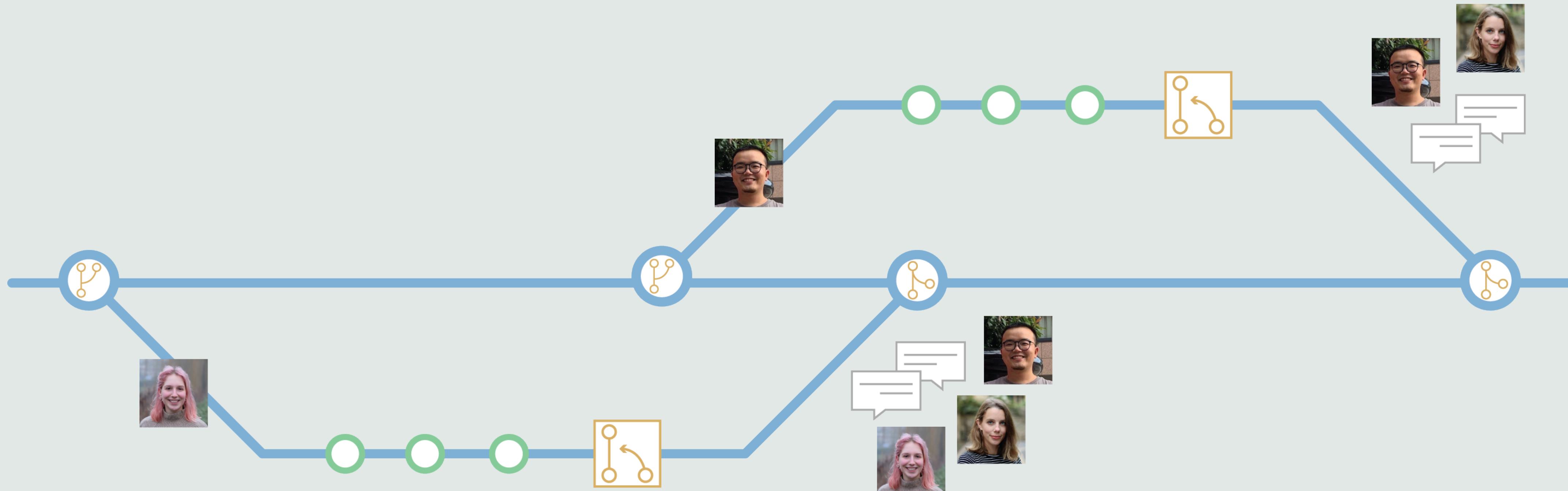
The screenshot shows a local development environment in VS Code. The Explorer sidebar shows a project structure for 'CODEADVENT2022' with folders for .github, input, and solutions, which further contain files like \_\_pycache\_\_, \_getinput.py, \_newday.py, day-01.py, day-02.py, and day-03.py (which is selected). The code editor displays the contents of day-03.py. The terminal shows the command 'zsh' and the path '(base) rosenasrawi@Roses-MacBook-Pro CodeAdvent2022 %'. Status bar details include 'Ln 1, Col 1', 'Spaces: 4', 'UTF-8', 'LF', and 'Python 3.8.5 ('base': conda)'.

```
1 # Day 3: Rucksack Reorganisation
2
3 from _getinput import *
4 import string
5
6 def split(sack):
7     m = len(sack)//2
8     return sack[:m], sack[m:]
9
10 def elfsupplies(rucksacks, part):
11
12     letters = list(string.ascii_letters)
13     mappings = {let: val+1 for val, let in enumerate(letters)}
14     total = 0; common = []
15
16     if part == 1:
17
18         for sack in rucksacks:
19             s1, s2 = split(sack)
```

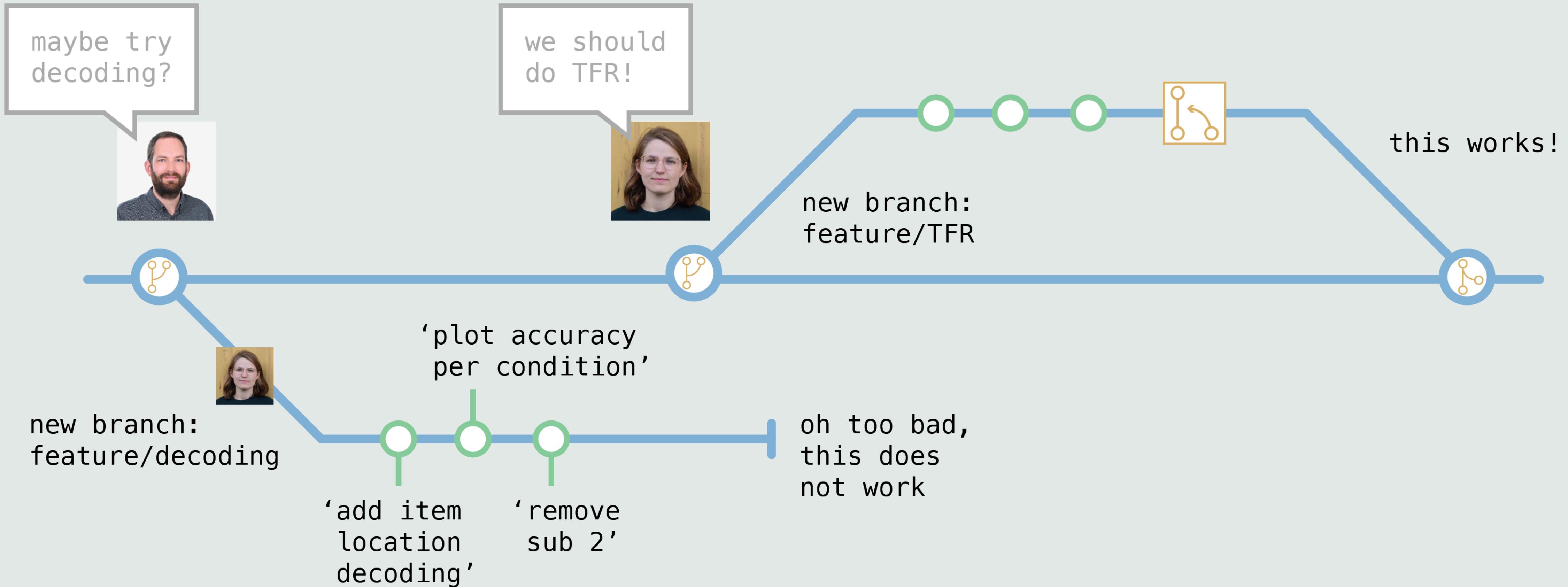
# GitHub | Collaborative



# GitHub | Collaborative

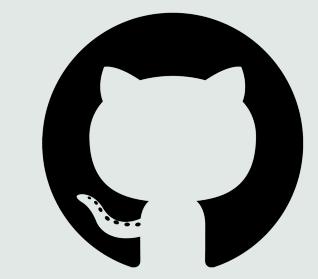


# GitHub | 'Solo' also useful!





**coffee!**



**GitHub**

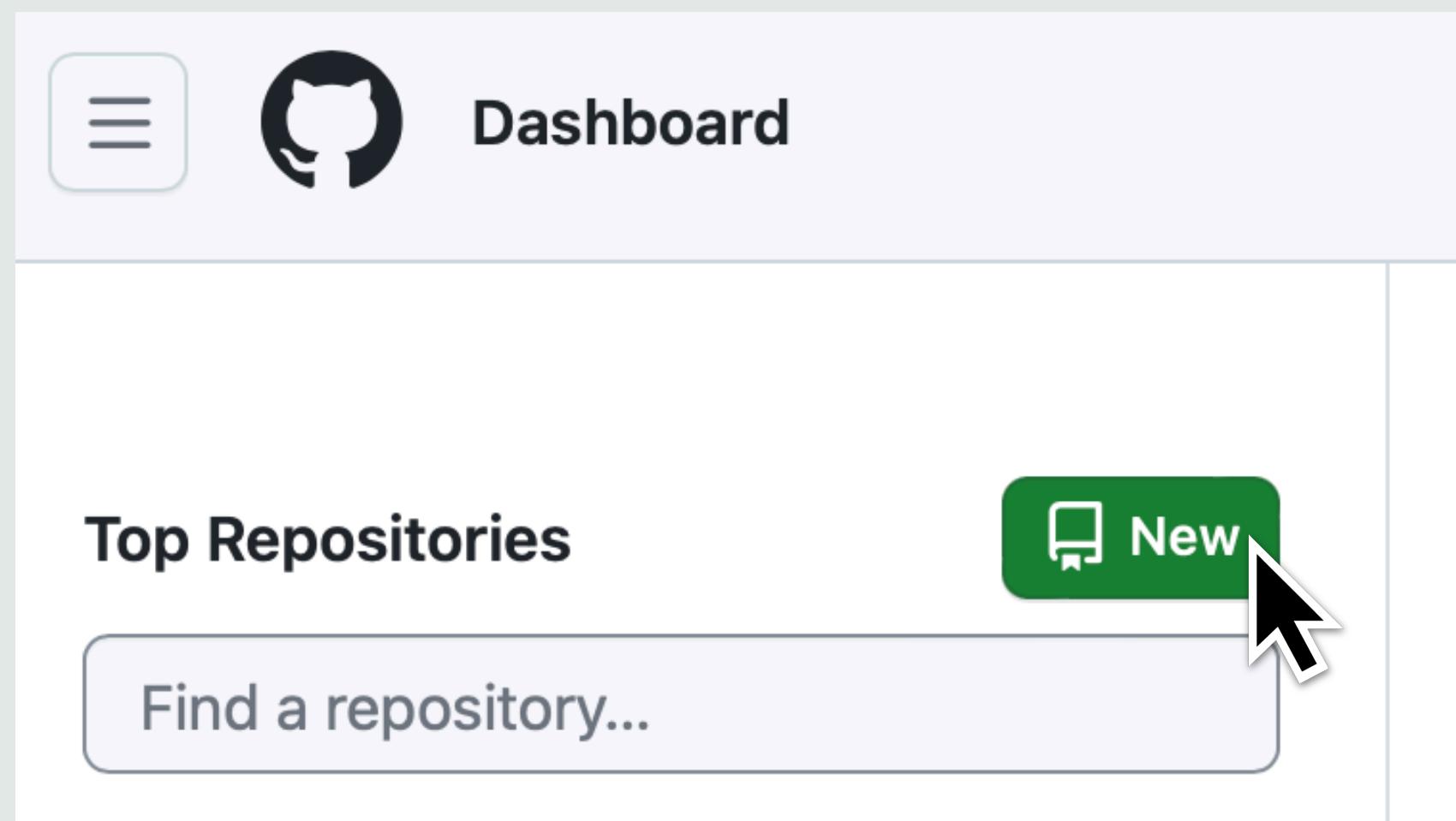
|

**getting started**



[www.github.com](https://www.github.com)

# GitHub | New repository



# GitHub | New repository

The GitHub Dashboard interface. It features a sidebar with a menu icon and the GitHub logo. The main area is titled "Dashboard". Below it, there's a section for "Top Repositories" and a search bar labeled "Find a repository...". A prominent green button with a clipboard icon and the word "New" is located in the center.

## Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository](#).

Required fields are marked with an asterisk (\*).

**Owner \*** **Repository name \***

rosenasrawi /   pbl-github-workshop is available.

Great repository names are short and memorable. Need inspiration? How about [super-duper-octo-parakeet](#)?

**Description (optional)**

An introduction to using git(hub)

**Public**  
Anyone on the internet can see this repository. You choose who can commit.

**Private**  
You choose who can see and commit to this repository.

**Initialize this repository with:**

**Add a README file**  
This is where you can write a long description for your project. [Learn more about READMEs](#).

**Add .gitignore**

.gitignore template: [None](#)

Choose which files not to track from a list of templates. [Learn more about ignoring files](#).

**Choose a license**

License: [None](#)

A license tells others what they can and can't do with your code. [Learn more about licenses](#).

This will set [main](#) as the default branch. Change the default name in your [settings](#).

You are creating a public repository in your personal account.

**Create repository**

# GitHub | cloning

The screenshot shows a GitHub repository page for "pbl-github-workshop" owned by "rosenasrawi". The repository is public and contains one branch ("main") and one file ("README.md"). A modal window is open over the page, specifically the "Code" dropdown menu, which provides options for cloning the repository using HTTPS, SSH, or GitHub CLI. The HTTPS URL is displayed as <https://github.com/rosenasrawi/pbl-github-workshop>. A large black cursor arrow points to the copy icon next to the URL.

Code dropdown menu options:

- Local
- Codespaces
- Clone
- HTTPS
- SSH
- GitHub CLI

Clone URL: <https://github.com/rosenasrawi/pbl-github-workshop>

Use Git or checkout with SVN using the web URL.

Readme

Activity

0 stars

1 watching

0 forks

An introduction to using git(hub)

Initial commit

Initial commit

README.md

pbl-github-workshop

An introduction to using git(hub)

© 2023 GitHub, Inc. Terms Privacy Security Status Docs Contact GitHub Pricing API Training Blog About

# GitHub | cloning

```
$ cd Directory/of/this/workshop
$ git clone [url-of-your-repo]

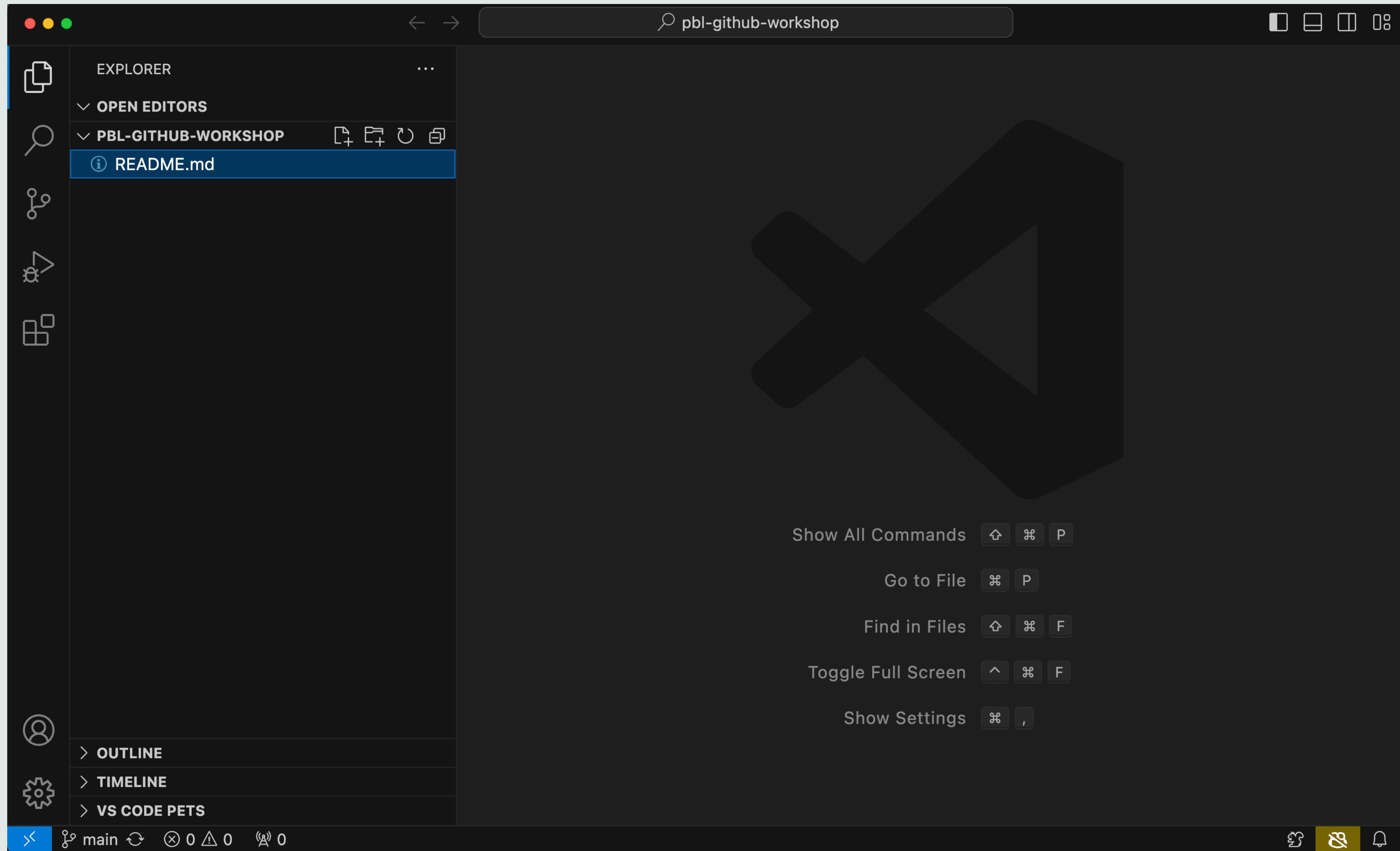
> Cloning into 'pbl-github-workshop'...
> remote: Enumerating objects: 3, done.
> remote: Counting objects: 100% (3/3), done.
> remote: Compressing objects: 100% (2/2), done.
> remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
> Receiving objects: 100% (3/3), done.
```



```
$ cd pbl-github-workshop
$ git status

> On branch main
> Your branch is up to date with 'origin/main'.

> nothing to commit, working tree clean
```



# GitHub | .gitignore

A screenshot of the Visual Studio Code (VS Code) interface. The title bar shows the repository name "pbl-github-workshop". The left sidebar (Explorer) lists files and folders: ".gitignore" (selected), ".DS\_Store", and "\_\_pycache\_\_" under the "OPEN EDITORS" section; ".gitignore" and "README.md" under the "PBL-GITHUB-WORKSHOP" section. The bottom left shows status icons: main\*, 0 changes, 0 conflicts, and 0 unstaged files. The bottom right shows file details: Line 2, Column 12, Spaces: 4, UTF-8, LF, Ignore, and a yellow status bar icon.

```
.gitignore
1 .DS_Store
2 __pycache__
```

# GitHub | .gitignore

The screenshot shows a dark-themed interface of the Visual Studio Code (VS Code) code editor. In the top right corner, there's a search bar containing the text "pbl-github-workshop". The Explorer sidebar on the left lists several items under "OPEN EDITORS" and "PBL-GITHUB-WORKSHOP". A ".gitignore" file is currently selected in the Explorer, showing its contents:

```
1 .DS_Store
2 __pycache__
```

In the bottom right corner, a terminal window displays the output of the command \$ git status:

```
$ git status

> On branch main
> Your branch is up to date with 'origin/main'.

> Untracked files:
  (use "git add <file>..." to include in what will be committed)
    .gitignore
```

The status bar at the bottom provides information about the current file ("main\*"), the number of changes (0), and the encoding ("UTF-8"). It also includes icons for ignore, GitHub, and notifications.

# GitHub | .gitignore

The screenshot shows the Visual Studio Code interface with the following details:

- Explorer View:** Shows two projects: "OPEN EDITORS" and "PBL-GITHUB-WORKSHOP". In "OPEN EDITORS", there are two files: ".gitignore" and "just-some-text.txt". In "PBL-GITHUB-WORKSHOP", there are two files: ".gitignore" and "just-some-text.txt".
- Editor View:** The "just-some-text.txt" file is currently open. Its content is:

```
Hello hello, this is just some text  
but it is just for me, I do not want  
to track this file
```
- Status Bar:** At the bottom, the status bar displays: "Ln 3, Col 19" and "Spaces: 4". It also shows icons for "main\*", "0 △ 0", "0", "UTF-8", "LF", "Plain Text", and a yellow gear icon.

# GitHub | .gitignore

The screenshot shows a dark-themed VS Code interface. In the Explorer sidebar, there are two open editors: '.gitignore' and 'just-some-text.txt'. The '.gitignore' file contains the following content:

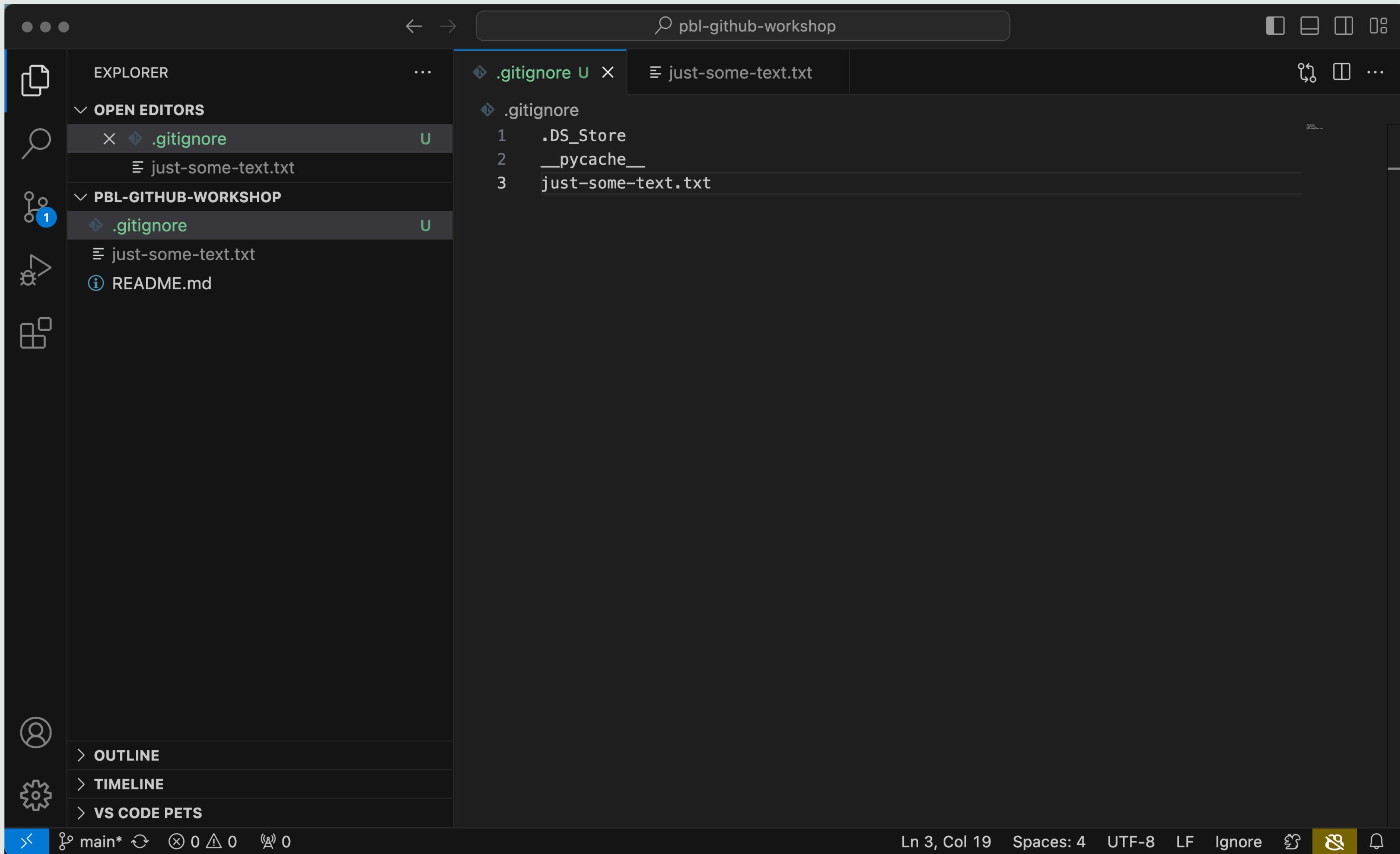
```
Hello hello, this is just some text  
but it is just for me, I do not want  
to track this file
```

The 'just-some-text.txt' file contains the text 'Hello hello, this is just some text but it is just for me, I do not want to track this file'. A modal window is displayed in the center, showing the output of the command '\$ git status':

```
$ git status  
  
> On branch main  
> Your branch is up to date with 'origin/main'.  
  
> Untracked files:  
  (use "git add <file>..." to include in what will be committed)  
  .gitignore  
  just-some-text.txt
```

The bottom status bar indicates the file is 'main\*' and has 0 changes.

# GitHub | .gitignore



A screenshot of the Visual Studio Code (VS Code) interface. The title bar shows "GitHub | .gitignore". The left sidebar has icons for Explorer, Search, Problems (with 1), and others. The Explorer sidebar shows a tree view with "OPEN EDITORS" containing ".gitignore" and "just-some-text.txt", and a folder "PBL-GITHUB-WORKSHOP" containing ".gitignore" and "just-some-text.txt", plus a "README.md" file. The main editor area displays the contents of a ".gitignore" file:

```
.gitignore
1 .DS_Store
2 __pycache__
3 just-some-text.txt
```

The status bar at the bottom shows "Ln 3, Col 19" and "Spaces: 4".

# GitHub | .gitignore

The screenshot shows a dark-themed instance of VS Code. In the Explorer sidebar, there are two entries under 'OPEN EDITORS': '.gitignore' and 'just-some-text.txt'. Below them is a folder named 'PBL-GITHUB-WORKSHOP' containing a '.gitignore' file and a 'just-some-text.txt' file. The 'just-some-text.txt' file has a small green 'U' icon next to it, indicating it is untracked.

The main editor area displays the contents of the '.gitignore' file:

```
.gitignore
1 .DS_Store
2 __pycache__
3 just-some-text.txt
```

A terminal window is open at the bottom, showing the output of the command \$ git status:

```
$ git status

> On branch main
> Your branch is up to date with 'origin/main'.

> Untracked files:
  (use "git add <file>..." to include in what will be committed)
    .gitignore
```

The status bar at the bottom indicates the current branch is 'main\*', and the file 'main\*' is the active one. It also shows statistics: 0 changes, 0 additions, 0 deletions, and 0 unstaged changes. The status bar also includes information about line and column counts, spaces, encoding, ignore settings, and icons for file operations like copy, paste, and refresh.

# GitHub | our first commit

The screenshot shows a dark-themed instance of VS Code. In the Explorer sidebar, there's a tree view with a root folder named "PBL-GITHUB-WORKSHOP". Inside this folder, there are two files: ".gitignore" and "just-some-text.txt". The ".gitignore" file contains three entries: ".DS\_Store", "\_\_pycache\_\_", and "just-some-text.txt". The "just-some-text.txt" file is shown as a preview. The status bar at the bottom indicates the current file is "main\*" and has 0 changes.

In the center, a terminal window displays the following command history:

```
$ git add .gitignore
$ git commit -m 'add .gitignore'

> [main c7d3dfa] add .gitignore
> 1 file changed, 3 insertions(+)
> create mode 100644 .gitignore
```

The status bar at the bottom of the terminal window shows "Ln 3, Col 19" and other commit details.



# print('Hello, world!')

A screenshot of the Visual Studio Code (VS Code) interface. The title bar shows the repository name "pbl-github-workshop". The Explorer sidebar on the left lists files and folders: ".gitignore", "hello-world.py", "just-some-text.txt", and a folder "PBL-GITHUB-WORKSHOP" containing ".gitignore" and "hello-world.py". The "hello-world.py" file in the Explorer is selected. The main editor area displays the code:

```
1 print('Hello, world!')
```

The status bar at the bottom shows the file is "main\*", has 0 changes (0 up, 1 down), and 0 unstaged changes. It also indicates the code is in Python 3.6.13 ('psychopy': conda) environment.

# GitHub | add, commit

REPO: pbl-github-workshop

EXPLORER

- .gitignore
- hello-world.py
- just-some-text.txt

PBL-GITHUB-WORKSHOP

- .gitignore
- hello-world.py
- just-some-text.txt
- README.md

```
$ git status  
$ git add hello-world.py  
$ git commit -m 'print hello world'
```

main\* ⌂ 0↓ 1↑ ⌂ 0 ⌂ 0 ⌂ 0

Ln 1, Col 23 Spaces: 4 UTF-8 LF { Python 3.6.13 ('psychopy': conda) ⌂ ⌂ ⌂

# GitHub | add, commit, push

The screenshot shows a dark-themed instance of the Visual Studio Code (VS Code) code editor. At the top, the title bar displays the GitHub logo and the repository name "pbl-github-workshop". The left sidebar contains the Explorer, Search, Problems, and Activity Bar sections. The Explorer section shows the project structure with files like ".gitignore", "hello-world.py", and "just-some-text.txt" under two different workspace roots: "OPEN EDITORS" and "PBL-GITHUB-WORKSHOP". The main editor area shows the "hello-world.py" file with the single line of code: `1 print('Hello, world!')`. A terminal window is open at the bottom, displaying the following output:

```
> On branch main
> Your branch is ahead of 'origin/main' by 2 commits.
  (use "git push" to publish your local commits)
> nothing to commit, working tree clean

$ git push
```

The status bar at the bottom provides information about the current file: "Ln 1, Col 23" and "Spaces: 4". It also indicates the file type as "Python" and the Python version as "3.6.13 ('psychopy': conda)".



# what happened remotely?

The screenshot shows a GitHub repository page for 'pbl-github-workshop'. The repository is public and has 1 branch and 0 tags. The main branch contains three commits:

- roenasrawi print hello world (c8376ee, 5 minutes ago, 3 commits)
- .gitignore (add .gitignore, 13 minutes ago)
- README.md (Initial commit, 2 days ago)

The README.md file content is:

```
pbl-github-workshop
An introduction to using git(hub)
```

The repository has 1 watch and 0 forks. It includes sections for About, Releases, Packages, and Languages.

**About**

An introduction to using git(hub)

**Readme**

**Activity**

**0 stars**

**1 watching**

**0 forks**

**Releases**

No releases published

[Create a new release](#)

**Packages**

No packages published

[Publish your first package](#)

**Languages**

Python 100%



# what happened remotely?

A screenshot of a GitHub repository page. The URL in the address bar is `github.com`. The repository name is `rosenasrawi / pbl-github-workshop`. The main navigation menu includes Code, Issues, Pull requests, Actions, Projects, Wiki, Security, Insights, and Settings. The `Code` tab is selected. The page title is **Commits**. A dropdown menu shows the current branch is `main`. The commit history is organized by date:

- Commits on Oct 20, 2023**
  - print hello world** by `rosenasrawi` committed 8 minutes ago. Includes copy, view, and diff icons.
  - add .gitignore** by `rosenasrawi` committed 16 minutes ago. Includes copy, view, and diff icons.
- Commits on Oct 18, 2023**
  - Initial commit** by `rosenasrawi` committed 2 days ago. Includes a **Verified** badge, copy, view, and diff icons.

At the bottom, there are **Newer** and **Older** buttons. The footer contains links to GitHub's Terms, Privacy, Security, Status, Docs, Contact GitHub, Pricing, API, Training, Blog, and About pages.



```
$ git branch  
> * main
```



```
$ git branch feature/goodbye  
  
$ git branch  
  
> feature/goodbye  
> * main
```



```
$ git checkout feature/goodbye  
> Switched to branch 'feature/goodbye'
```



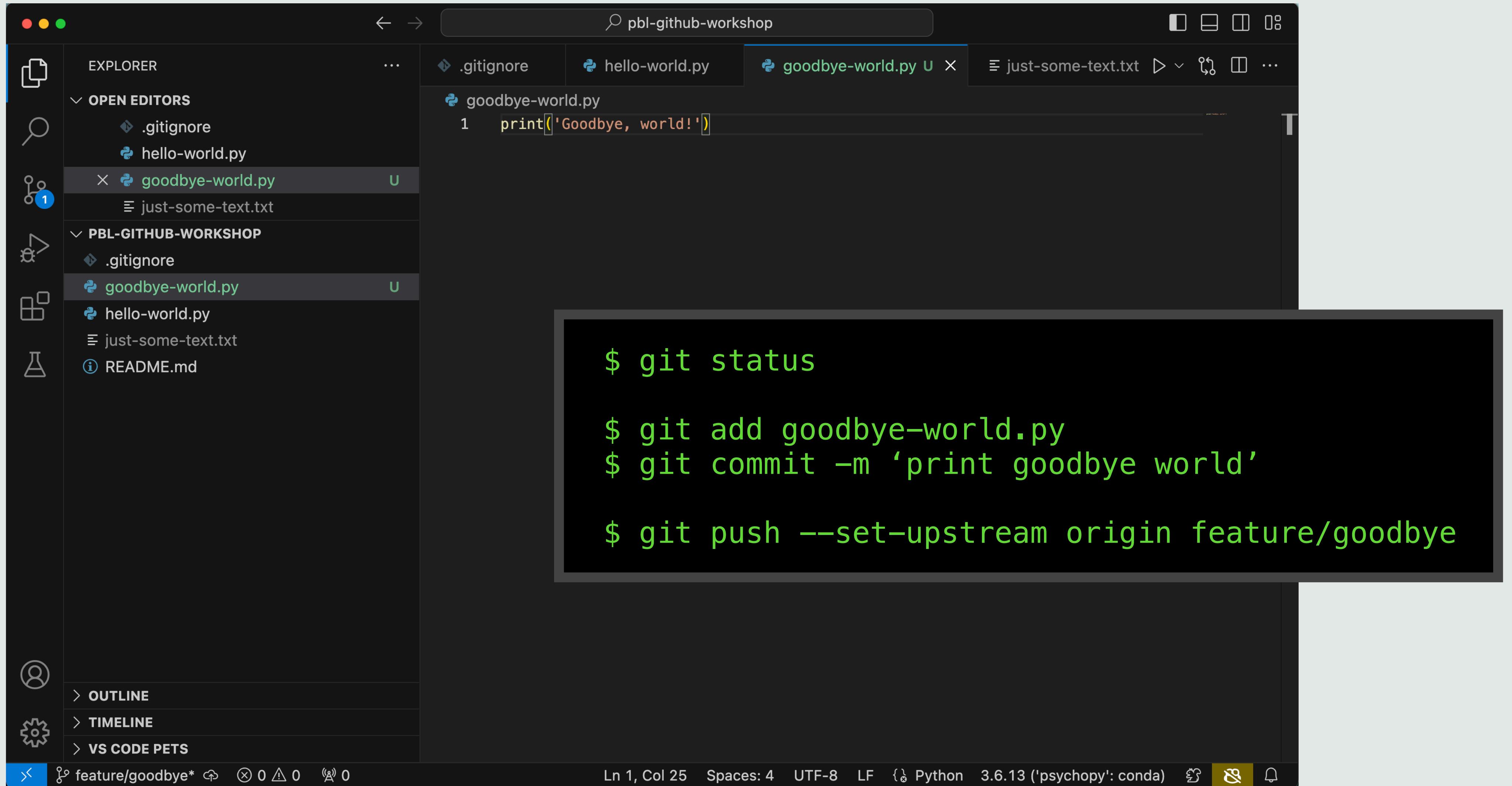
# print('Goodbye, world!')

The screenshot shows a dark-themed instance of the Visual Studio Code (VS Code) code editor. The title bar displays the repository name "pbl-github-workshop". The Explorer sidebar on the left lists several files and folders, including ".gitignore", "hello-world.py", "goodbye-world.py" (which is currently selected), and "just-some-text.txt". In the center, the code editor pane contains the file "goodbye-world.py" with the single line of code:

```
1 print('Goodbye, world!')
```

The status bar at the bottom provides information about the file: "feature/goodbye\* 25 0 0 0", "Ln 1, Col 25", "Spaces: 4", "UTF-8 LF", "Python 3.6.13 ('psychopy': conda)", and icons for file operations like save and close.

# GitHub | add, commit, push *to branch*



The screenshot shows a dark-themed instance of VS Code. In the Explorer sidebar, there are two expanded sections: 'OPEN EDITORS' containing '.gitignore', 'hello-world.py', and 'goodbye-world.py' (which is currently selected), and 'PBL-GITHUB-WORKSHOP' containing '.gitignore', 'goodbye-world.py' (selected), 'hello-world.py', 'just-some-text.txt', and 'README.md'. The 'goodbye-world.py' editor shows the single-line code: `print('Goodbye, world!')`. A terminal window in the bottom right displays the following command sequence:

```
$ git status  
  
$ git add goodbye-world.py  
$ git commit -m 'print goodbye world'  
  
$ git push --set-upstream origin feature/goodbye
```

The status bar at the bottom indicates the file is 'feature/goodbye\*' and shows line 1, column 25, spaces: 4, encoding: UTF-8, LF, Python 3.6.13 ('psychopy': conda). There are also icons for file operations like close, save, and refresh.



# what happened remotely?

The screenshot shows a GitHub repository page for 'pbl-github-workshop'. The repository is public and has 1 branch and 0 tags. It contains four commits from user 'rosenasrawi': 'print hello world' (c8376ee, 29 minutes ago), 'add .gitignore' (37 minutes ago), 'Initial commit' (2 days ago), and 'print hello world' (29 minutes ago). The README.md file contains the text 'An introduction to using git(hub)'. The repository has 1 watch, 0 forks, and 0 stars. The 'About' section describes it as 'An introduction to using git(hub)'.

**feature/goodbye** had recent pushes less than a minute ago

**Compare & pull request**

main ▾ 1 branch 0 tags

Go to file Add file ▾ Code ▾

**rosenasrawi** print hello world c8376ee 29 minutes ago 3 commits

.gitignore add .gitignore 37 minutes ago

README.md Initial commit 2 days ago

hello-world.py print hello world 29 minutes ago

**README.md**

**pbl-github-workshop**

An introduction to using git(hub)

**About**

An introduction to using git(hub)

Readme

Activity

0 stars

1 watching

0 forks

**Releases**

No releases published

[Create a new release](#)

**Packages**

No packages published

[Publish your first package](#)

**Languages**

Python 100%

# GitHub | our branches differ! pull request?

A screenshot of a GitHub repository page for `pbl-github-workshop`. The repository is public and has 3 commits from `rosenasrawi`. A yellow banner at the top indicates that the `feature/goodbye` branch has recent pushes less than a minute ago. A green button labeled `Compare & pull request` is highlighted with a mouse cursor. The repository details show 0 stars, 1 watching, and 0 forks. Sections for `Releases`, `Packages`, and `Languages` are also visible.

feature/goodbye had recent pushes less than a minute ago

Compare & pull request

rosenasrawi print hello world c8376ee 29 minutes ago 3 commits

.gitignore add .gitignore 37 minutes ago

README.md Initial commit 2 days ago

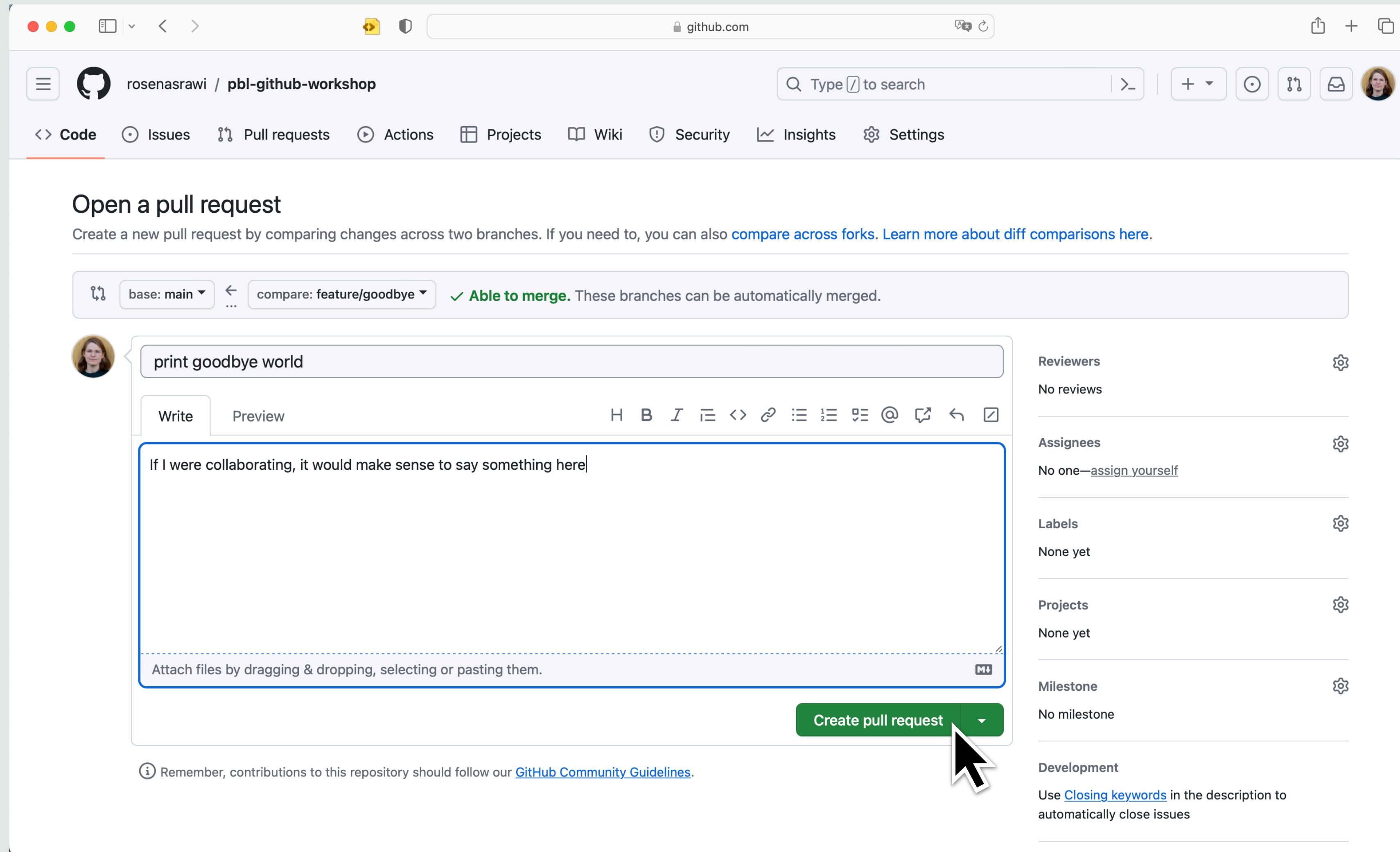
hello-world.py print hello world 29 minutes ago

**Releases**  
No releases published [Create a new release](#)

**Packages**  
No packages published [Publish your first package](#)

**Languages**  
Python 100%

# GitHub | pull request



A screenshot of a GitHub repository titled "rosenasrawi / pbl-github-workshop". The "Code" tab is selected. A modal window titled "Open a pull request" is open, showing a comparison between "base: main" and "compare: feature/goodbye". The message field contains "print goodbye world" and "If I were collaborating, it would make sense to say something here". A cursor points at the "Create pull request" button. The right side of the screen shows settings for reviewers, assignees, labels, projects, milestones, and development.

base: main ▾ ← compare: feature/goodbye ✓ Able to merge. These branches can be automatically merged.

print goodbye world

If I were collaborating, it would make sense to say something here|

Attach files by dragging & dropping, selecting or pasting them.

Create pull request ▾

Remember, contributions to this repository should follow our [GitHub Community Guidelines](#).

Reviewers: No reviews

Assignees: No one—[assign yourself](#)

Labels: None yet

Projects: None yet

Milestone: No milestone

Development: Use [Closing keywords](#) in the description to automatically close issues

# GitHub | merge pull request

A screenshot of a GitHub pull request page. The URL in the address bar is `github.com`. The navigation bar includes links for Code, Issues, Pull requests (1), Actions, Projects, Wiki, Security, Insights, and Settings. The main title of the pull request is "print goodbye world #1". A green "Open" button is visible. Below it, a message from user `rosenasrawi` says: "If I were collaborating, it would make sense to say something here" followed by a smiley face emoji. The commit hash is `60d7bbc`. A note at the bottom says "Add more commits by pushing to the `feature/goodbye` branch on `rosenasrawi/pbl-github-workshop`". On the right side, there are sections for Reviewers, Assignees, Labels, Projects, Milestone, and Development, all currently set to "None yet". At the bottom left, a large green "Merge pull request" button has a black cursor arrow pointing to its center. Below the button, text reads "You can also open this in GitHub Desktop or view command line instructions."



# remote: delete feature/goodbye

A screenshot of a GitHub pull request merge page. At the top, a purple banner indicates the merge was successful: "Merged rosenasrawi merged 1 commit into main from feature/goodbye now". Below the banner, the commit history shows a single commit from "print goodbye world" (60d7bbc) and a merge commit from "rosenasrawi merged commit 986f206 into main now". A tooltip message "Pull request successfully merged and closed" appears over the merge commit, stating "You're all set—the feature/goodbye branch can be safely deleted." A cursor is hovering over the "Delete branch" button next to this message. To the right of the commit history, there are sections for Reviewers (No reviews), Assignees (No one—assign yourself), Labels (None yet), Projects (None yet), Milestone (No milestone), Development (Successfully merging this pull request may close these issues. None yet), and Notifications (Customize, Unsubscribe). At the bottom, a note says "Remember, contributions to this repository should follow our [GitHub Community Guidelines](#)." A pro tip at the bottom suggests adding comments to specific lines under "Files changed".

# GitHub | local: back to main

```
$ git checkout main
$ git branch

>   feature/goodbye
> * main
```

# GitHub | local: pull (!!!)

```
$ git pull

> remote: Enumerating objects: 1, done.
> remote: Counting objects: 100% (1/1), done.
> remote: Total 1 (delta 0), reused 0 (delta 0), pack-reused 0
> Unpacking objects: 100% (1/1), 637 bytes | 637.00 KiB/s, done.
> From https://github.com/rosenasrawi/pbl-github-workshop
      c8376ee..986f206  main            -> origin/main
> Updating c8376ee..986f206
> Fast-forward
    goodbye-world.py | 1 +
    1 file changed, 1 insertion(+)
    create mode 100644 goodbye-world.py
```

# GitHub | local: delete feature/goodbye

```
$ git branch -D feature/goodbye  
> Deleted branch feature/goodbye (was 60d7bbc).  
  
$ git branch  
* main
```

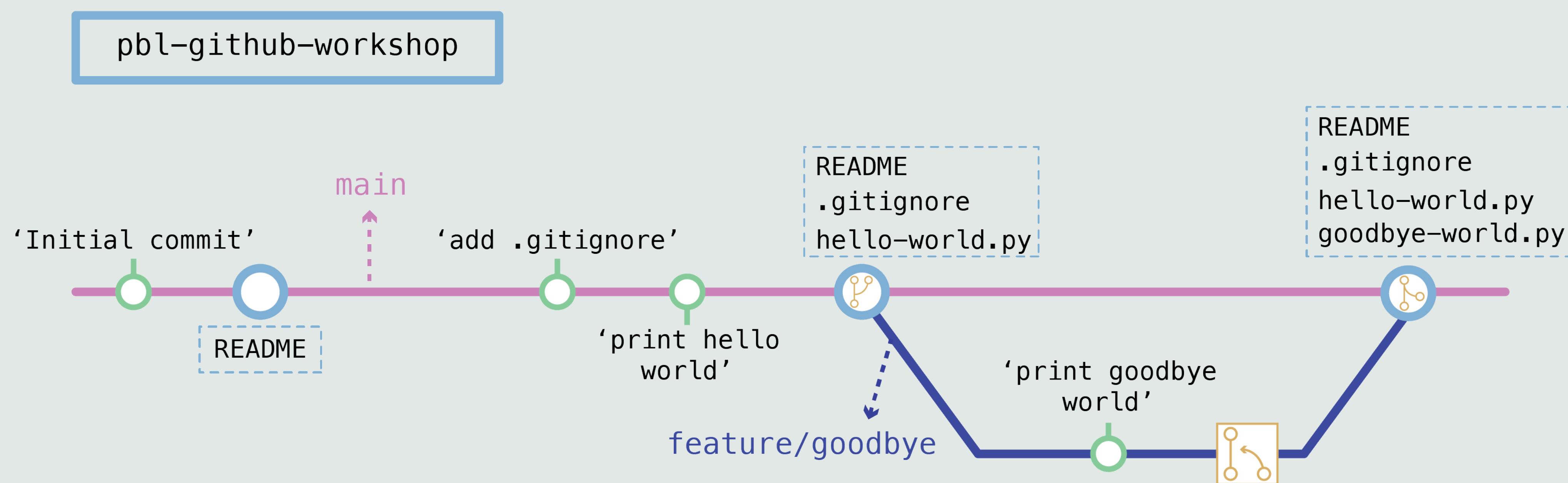
# GitHub | local: back on track!

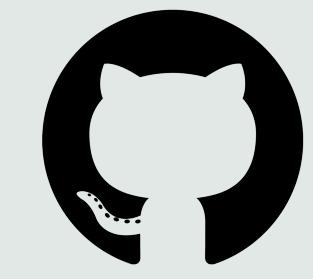
The screenshot shows a dark-themed instance of VS Code. In the Explorer sidebar, there are two expanded sections: 'OPEN EDITORS' and 'PBL-GITHUB-WORKSHOP'. Under 'OPEN EDITORS', files '.gitignore', 'hello-world.py', and 'goodbye-world.py' are listed; 'goodbye-world.py' is currently selected and its content is visible in the editor. Under 'PBL-GITHUB-WORKSHOP', files '.gitignore', 'goodbye-world.py', and 'hello-world.py' are listed; 'hello-world.py' is currently selected. Other files like 'just-some-text.txt' and 'README.md' are also present. The terminal window at the bottom displays the output of the command '\$ git status', which indicates the repository is up-to-date and clean.

```
$ git status
On branch main
Your branch is up to date with 'origin/main'.
nothing to commit, working tree clean
```

VS Code status bar: Ln 1, Col 25 Spaces: 4 UTF-8 LF Python 3.6.13 ('psychopy': conda)

# GitHub | recap: what did we do?





**GitHub** | **bonus assignment**



Oh no, we've changed our mind!

We no longer find it useful to have two separate scripts to print 'Hello, world!' and 'Goodbye, world!'. Instead we want just one script, in which we print one or the other based on the situation.

If it's between 9.00 and 17.00, we want it to `print('Hello, world!')`, else it should `print('Goodbye, world!')`

- (1) Create a new branch **feature/hello-goodbye**
- (2) Change your **working branch** to this one
- (3) **Delete** the script `goodbye-world.py`
- (4) **Update** the script `hello-world.py` as described above.
- (5) **Add** and **commit** your changes, then **push** your commits
- (6) Create a **pull request**, and **merge** your branches, **delete** your branch remotely
- (7) Checkout main, **pull remote changes**, then **delete** your branch locally

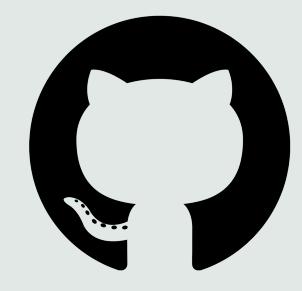
## HINTS

### Python

```
from datetime import datetime  
  
.now()  
.strftime()  
int()
```

### Git

```
$ git status  
  
$ git branch  
$ git branch [name]  
$ git checkout [name]  
  
$ git add/rm [file]  
$ git commit -m 'blablabla'  
  
$ git push --set-upstream origin [branch]  
$ git branch -D [name]
```



**GitHub** | **easy, right?**